DOCUMENT RESUME

ED 201 643 TM 810 160

AUTHOR Klein-Braley, Christine
TITLE Test Analysis With SPSS.
INSTITUTION Interuniversitare Sprachtestgruppe (West Germany).
REPORT NO IUS-WP-2
PUB DATE Nov 79
NOTE 62p.

EDRS PRICE MF01/PC03 Plus Postage.
DESCRIPTORS *Computer Programs; *Programming; *Statistical Analysis; *Tests
IDENTIFIERS *Statistical Package for the Social Sciences

ABSTRACT

This is the revised version of a paper originally produced for the AKS Test Teach-In held at the Gesamthochschule in Duisburg, West Germany, in March 1979. Details about computing are expanded in the revised version, but no attempt has been made to go into the intricacies of test statistics. This paper assumes that the reader knows what statistics are needed and how to interpret them. How to obtain relevant information about tests using a computer and the Statistical Package for the Social Sciences (SPSS) is demonstrated. SPSS is a "shorthand" method of using the computer to produce statistical analyses of data without having to learn one of the more flexible, but infinitely more complicated, programming languages. Topics covered in this guide include: (1) layout of an SPSS program; (2) program input--data; (3) SPSS syntax; (4) first steps in programming--data definition; (5) comfort for the user--labels; (6) let the computer do the work--data transformation; (7) entering data; (8) statistical procedures in SPSS; (9) frequencies; (10) reliability; (11) Pearson product-moment correlation; (12) output from program BREAKDOWN; (13) t-test; (14) breakdown for lists; (15) finish; (16) edit; and (17) error routines. (RL)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☑ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

RECEIVED
ERIC/TM

ED201643

WORKPAPERS

IUS INTER UNIVERSITÄRE SPRACHTESTGRUPPE

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
C. Klein-Braley
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

CO-CHAIRMEN OF IUS
CHRISTINE KLEIN-BRALEY
GESAMTHOCHSCHULE
DUISBURG · FACHBER 3
LOTHARSTRASSE 65
4100 DUISBURG 1
0203/305-395

DOUGLAS K. STEVENSON
UNIVERSITÄT ESSEN
FACHBEREICH 3
ZIMMER R12 R04 A14
4300 ESSEN 1
0201/183-3462

TEST CONSULTANT
ULRICH RAATZ
GESAMTHOCHSCHULE
DUISBURG · FACHBER 2
LOTHARSTRASSE 65
4100 DUISBURG 1
0203/305-552

TM 810 160

The IUS Workpapers are published at Essen and Duisburg, Federal Republic of Germany, by the Interuniversitäre Sprachtestgruppe in cooperation with the Universität Essen and the Universität Duisburg. *Unless otherwise indicated, all IUS Workpapers may be copied or quoted as long as the source is acknowledged and no financial profit is involved.*

The IUS is especially grateful to the Universität Essen for its consistent and generous support of the Workpapers through research funding, and to Herr Wilken, Director of the University Printing Office, for his advice and support.

The Interuniversitäre Sprachtestgruppe welcomes manuscripts which treat topics of more than temporary interest to language testers working in educational linguistics and language pedagogy. Those interested in submitting manuscripts should send three typewritten copies (written in English, French, or German) to either of the IUS Co-chairmen, Christine Klein-Braley or Douglas K. Stevenson.

IUS WORKPAPERS 2

2

TEST ANALYSIS WITH SPSS

Christine
Klein-Braley

November 1979

Author's Address:

Christine Klein-Braley, M.A.
Fachbereich 3 - Anglistik
Universität Duisburg
Lotharstraße 65
4100 Duisburg 1

Essen and Duisburg:
Interuniversitäre
Sprachtestgruppe

TABLE OF CONTENTS

	PAGE
0. Introduction	iii
1. SPSS: Statistical Package for the Social Sciences	1
Reference books	1
2. Layout of an SPSS program	2
3. Program input: data	2
Variables and values	3
Demographic and test data	3
Punch card	4
Suitable point at which to attempt Exercise 1	
4. SPSS syntax	6
Variable names	6
Commands and specifications	6
Special characters	7
Commas and blanks	8
The keyword TO	8
5. First steps in programming: data definition	9
RUN NAME	10
DATA LIST	10
INPUT MEDIUM	11
N OF CASES	11
MISSING VALUES	11
Suitable point at which to attempt Exercise 2	
6. Comfort for the user: labels	13
VAR LABELS	13
VALUE LABELS	14
Deck placement of VAR LABELS and VALUE LABELS	15
Suitable point at which to attempt Exercise 3	
7. Let the computer do the work: data transformation	16
COMPUTE	16
COUNT	17
Missing data in data transformations	18
ASSIGN MISSING	19
Suitable point at which to attempt Exercise 4	
8. Entering data	20
READ INPUT DATA	20
Summary of cards discussed so far	21
9. Statistical procedures in SPSS	22
Table of test statistics and SPSS procedures	23
Complete program for exhibit 4	24
10. FREQUENCIES	25
Output from program FREQUENCIES	26

TABLE OF CONTENTS (cont.)

PAGE

11. RELIABILITY 27
Output from program RELIABILITY: items 29
Output from program RELIABILITY: test 30

12. PEARSON CORR 31
Output from program PEARSON CORR 31

13. BREAKDOWN 31
Output from program BREAKDOWN 32

14. T-TEST 33

15. BREAKDOWN for lists 33

16. FINISH 33

17. EDIT 34

18. Error routines 35
Write cases 35

19. A final word 36

APPENDIX

Exhibit 1 37
Exhibit 2 38
Exhibit 3 39
Exhibit 4 40

Exercise 1 41
Exercise 2 42
Exercise 3 43
Exercise 4 44

Solution to Exercise 1 45
Solution to Exercise 2 46
Solution to Exercise 3 47
Solution to Exercise 4 48

Complete solution to Exhibit 3 49

0. Introduction

This is the revised version of a paper originally produced for the AKS Test Teach-In held at the Gesamthochschule in Duisburg in March 1979. At the Test Teach-In the participants spent the first day learning about the statistical aspects of language testing with Prof. Dr. Ulrich Raatz, the second day learning about the history of language testing, language testing theory and the problems of validation, with Dr. Douglas K. Stevenson[1], and only on the third day did they join my fellow-tutor, Dipl. Inf. Hans-Dieter Weckmann, and myself for the computing. In the revision of the script I have tried to expand details about computing where they may be unclear to a beginner in the field, however I have made no attempt to go into the intricacies of test statistics: this paper assumes the reader knows what statistics he needs and how to interpret them. My aim here is to demonstrate how to obtain the relevant information about tests using a computer.

The most effective way of learning how to use a computer is to let the computer teach you. The SPSS EDIT facility (see below, p. 35) will accept incomplete programs (provided you add each bit to the previous bits, i.e. build up the program in sequential order), it will check all the punch cards for typing, careless, logical and syntactical errors, and report what it has found. Consequently it would be ideal if, after working through this paper once, you tackled the exercises at the appropriate points, punched the necessary cards, and sent them through the computer (using EDIT as the first SPSS card every time). For those people who do not have easy access to the computer I have provided one correct solution to the exercises. Any other solution which emerges from the computer accompanied by the remark:

NUMBER OF ERRORS DETECTED 0

is equally correct.

My thanks are due to Prof. Dr. Helmut E. Lück who introduced me to SPSS and guided my first steps in testing and in computing, to Hans-Dieter Weckmann and CYBER 76 for checking my SPSS for this paper, especially, however, to Penelope and Michael Sparrow, who read through the original manuscript looking for muddled expressions. Any confusion that remains in the manuscript is therefore all my own work, and I would be most grateful to be told about it.

Duisburg, November 1979 Christine Klein-Braley

---

1) IUS Workpaper Number 1: Douglas K. Stevenson: SELECTED BASIC RESOURCES IN FOREIGN LANGUAGE TESTING, June, 1979. Revised September, 1980.

1. SPSS: Statistical Package for the Social Sciences

SPSS is a "shorthand" method of using the computer to produce statistical analyses of data without having to learn one of the more flexible but infinitely more complicated programming languages (e.g. FORTRAN, ALGOL, COBOL).

Reference Books:

A. Nie, Bent, Hull et al.
SPSS: Statistical Package for the Social Sciences
2nd Edition
McGraw-Hill, New York, 1975

B. Beutel, Küffner, Röch et al.
SPSS - Statistik-Programm für die Sozialwissenschaften
2. Auflage
Gustav Fischer Verlag, Stuttgart, 1978

C. Informationsschriften from your local computing centre.

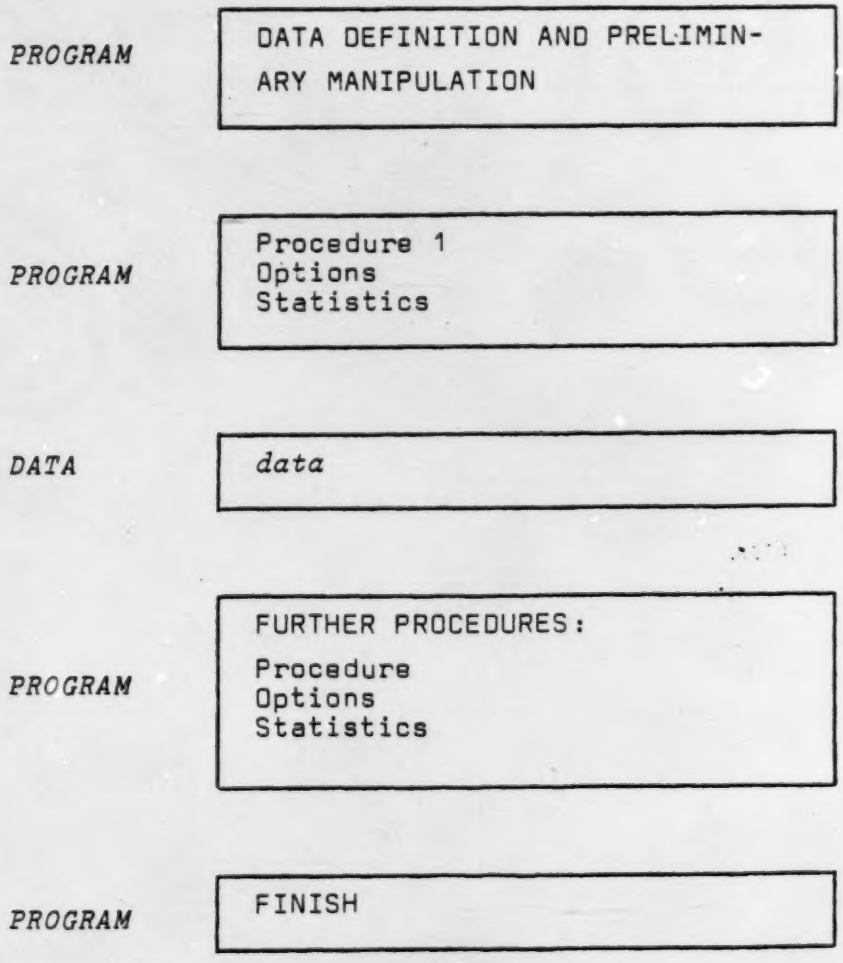
The McGraw-Hill manual is so clearly written and so explicit that it is well worth buying if you decide that you enjoy computing. Beutel, while cheaper, is much shorter, consequently more compressed, and - I find - in parts mysterious. It is however very useful for reference once the basic principles of SPSS have been grasped.

SPSS looks like English, and those parts of the program which are fixed use English words. However there is no reason why the parts of the program which are defined by the user should not be in German or in any other language.

2. Layout of an SPSS program

In order to make a computer function it is necessary to tell it what it has to do (PROGRAM) and then to give it material on which it can carry out the instructions (DATA). Both program and data are fed to the machine in the form of punch cards.

Any SPSS program will consist of the following blocks:



3. Program input: data

SPSS analyses data which are organised in the form of cases. Cases can be people (as in our tests), institutions (for instance different schools, universities, churches etc.)

or anything else where the same discrete units of information have been collected for a number of different objects. In tests each person is a case, the data are the information available about the person. In tests we can distinguish between demographic data: age, sex, qualification etc. and test data: the responses to test items.

Let us assume that we have administered a 20 item test to our students and have asked them to specify name, age, sex, Semesterzahl. 12 students took the test. Normally, of course, it would not be economical to use a computer for such a small group, but we will keep the number small for demonstration purposes. The raw data are listed on Exhibit 1. We have 4 demographic variables (name, age, sex, Semesterzahl) and 20 test variables (the answers to the test questions). Each question we ask is a variable: we ask each student the same questions. However the students do not all give the same answers. For instance some are male and some are female. The possible different answers to the same question are called the values of that variable. Obviously each person can have only one value for each variable: either male or female for sex; either right or wrong for each test variable (item).

SPSS prefers to receive data in the form of numbers; there are very few procedures for dealing with alphabetic (alphanumeric) variables. Consequently we assign numbers to the individual values:

DEMOGRAPHIC DATA

Name	a serial identification number (ID); 1 to 12
Age	is already available as a number. If no details are given, age = 99
Sex	male = 1
	female = 2
	if no details are given, sex = 9
Semester	is already available as a number. If no details are given, Semester = 9

TEST DATA

Item 1 to Item 20 right = 1
wrong = 0
not attempted = 9

In the demographic variables the numbers merely serve the function of names, so that there is no logical reason why 1 should be male and 2 female rather than the other way round. However in the test data there are good reasons for coding a correct answer as 1 since here 1 equals one point to be added to the test score.

Punch cards

The computer reads the data from punch cards:

and the holes are made (by you!) in the card with a card punch. What the computer actually reads are the holes; the text along the top edge of the card is for the convenience of the human being involved.

Each variable is punched in a specific fixed position (column) on the card, so that the computer can be told: "Read AGE from columns 5 and 6; read SEX from column 18" etc. For each variable we must determine the number of columns

necessary, e.g. a four figure number will need four columns, a one figure number only one. We can use as many punch cards as we need to contain the data for one case; however if we need more than one card it is sensible to repeat the ID number at the beginning of each new card and to number the cards through. All 80 card columns can be used for data.

For demonstration purposes we will use two cards for the data of Exhibit 1, though everything would fit on one card. Card 1 will hold the demographic variables; card 2 the test variables. The ID code will occupy 3 columns (cols 1 - 3) of each card; this allows for up to 999 test subjects. So the coding plan would be:

CARD	VARIABLE	COLUMNS	CODE
1	ID	1 - 3	001 to N (the number of the last subject)
	CN (card number)	4	1
	AGE	5 - 6	as given or 99
	SEX	7	1, 2 or 9
	SEM (Semesterzahl)	8	as given or 9

CARD	VARIABLE	COLUMNS	CODE
2	ID 2	1 - 3	001 to N
	CN 2	4	2
	ITEM 1 to ITEM 20	5 - 24	0, 1 or 9

Now all we have to do is to prepare this information for the computer. This brings us to programming proper.

4. SPSS syntax

Like other languages computer languages have their own syntax. However, unlike natural languages, computer languages have completely regular syntax. SPSS Syntax conforms to the following rules:

A. All variables used in a program must be named (any names you like!); each name must begin with a letter of the alphabet (but can contain numbers), must not have more than 8 characters (letters or numbers), must not contain any characters other than letters or numbers.

EXAMPLES OF VALID SPSS NAMES:
SEX, AGE, OCCUP, SEMZAHL, ITEM1, ITEM2, ITEM3A etc.

EXAMPLES OF INVALID SPSS NAMES:
OCCUPATION (has 10 characters, not 8 or less)
A+B (+ is neither a letter nor a number)
2ITEM (begins with a number)
JO JO (a space for the computer is a character; SPSS names must consist only of letters and numbers!)

B. Each command (instruction to the computer to initiate a new statistical procedure) begins in column 1 of a punch card and must be correctly spelled. The specifications (details of the variables to be used) to the command must not begin before column 16. However they can begin after column 16 of the same card, or even on column 16 of the following card.
Specifications can be continued from one physical punch card to the next, but always begin at or after column 16. Only commands may begin in column 1.

EXAMPLE:

1 16

FREQUENCIES*	GENERAL = SEX AGE	*card 1*
	ITEM1 ITEM2	*card 2*
	ITEM3	*card 3*

Variable names may not be split between cards when going on from one card to the next, though labels (see below, p. 13) may be split.

C. Special characters must be used in certain parts of the program and must not be used anywhere else at all.

() Parentheses are used to specify values of a variable, to specify conditions on IF and RECODE cards and in certain other places. (See below, p.34).

/ is used to indicate to the computer that information about one unit is finished and that the next unit is starting.

EXAMPLE:

1 16

MISSING VALUES AGE(99)/SEX(9)/ITEM1 TO ITEM20 (9)

Here the computer is told that if AGE is unknown (i.e. missing) 99 has been punched, if SEX is unknown 9 has been punched, if the answer to any item between ITEM1 and ITEM20 is missing, 9 has been punched.

---

* The name of the command is FREQUENCIES. This tells the computer it is to count. In the specification field (everything other than the command FREQUENCIES) the computer is first told that it is to use the GENERAL method of counting, and that it is to perform the counting on the variables SEX, AGE, ITEM1, ITEM2 and ITEM3.

The other special characters are generally used in arithmetical expressions:

+	plus
-	minus
*	multiply
**	raised to the power of (**2 = squared, **3 = cubed)
/	divided by
=	equals

EXAMPLE:

To make SPSS perform computations on the variables A, B, C and D to produce a 5th variable E according to the formula

$$e = \frac{a \times (b - c)}{d^2}$$

you would write:

```
1               16
COMPUTE         (A * (B - C) ) / (D**2)
```

The operations in parentheses are carried out first, then multiplication and division, finally addition and subtraction.

D. Commas and blanks can be used interchangeably. At least one blank or comma must be used between each discrete unit of information, e.g. you must write either AGE,SEX or AGE SEX orAGE, SEX but notAGESEX. The computer will reject AGESEX since it has not met a variable with this name before. However there need be no blanks or commas between special characters and other elements. (See example under C above). You may use as many blanks or commas as you want or need; except in labels (See below, p.13) the computer will ignore them.

E. The keyword TO

When three or more adjacent variables are to be referred to anywhere in the program (except on the VAR LABELS card, see below, p.13) only the first and last elements need to be mentioned if TO is used.

EXAMPLE:

```
1               16
FREQUENCIES     GENERAL = ITEM1 TO ITEM20
```

Here FREQUENCIES will be produced for all items from 1 to 20, not only for the ones mentioned here. This is an extremely economical procedure!

5. First steps in programming: data definition

If you look back at page 2 you will see that the first block of program is called DATA DEFINITION AND PRELIMINARY MANIPULATION. The data have to be defined for the computer because what you are entering are your data, and the only way for the computer to find out how the data are structured is for you to tell it. In other words it needs to know:

INFORMATION NEEDED	CONTROL CARD USED
a) what you want to call your program	RUN NAME
b) how many cards you have used for the data	DATA LIST
c) the names you have used for the variables, which columns you have punched the variables in and the number of columns needed	DATA LIST
d) how you are going to read the data into the computer	INPUT MEDIUM
e) how many cases you have	N OF CASES
f) which numbers you have used for missing values of variables	MISSING VALUES

The name of the control card is a command. In the data definition block of the program the control cards are used to give the computer information rather than to tell it to carry out an order. Consequently the names (or commands) for the individual control cards are always punched beginning from column 1 and they must be spelled exactly as they are here (or in the various manuals). The specification field of the control card begins from column 16 and is provided by the user according to the way his data are structured.

The data definition block for the program to analyse the data of exhibit 1 would look like this:

COMMAND	SPECIFICATION
1	16
RUN NAME	ANY NAME YOU LIKE: YOUR CHOICE
DATA LIST	FIXED (2)
	/1 ID1 1-3 CN1 4 AGE 5-6 SEX 7 SEM 8
	/2 ID2 1-3 CN2 4 V1 TO V20 5 - 24
INPUT MEDIUM	CARD
N OF CASES	12
MISSING VALUES	AGE (99) / SEX SEM V1 TO V20 (9)

Explanation of the individual commands:

A. RUN NAME must be present in the card deck (i.e. the complete, correctly arranged, undamaged set of cards you read into the computer). The name can fill the rest of one card from columns 16 to 80. The run name will be printed on every page of output (the printed list produced by the computer).

B. DATA LIST fulfils a variety of functions:

FIXED = the data are punched in fixed columns

(2) = there are 2 cards per case

/1 = go to card 1

ID1 1-3 = the variable which I have decided to call ID1 has been punched in columns 1 to 3

CN1 4 = the variable I have decided to call CN1 has been punched in column 4

AGE 5-6 = AGE is punched in columns 5 to 6 (i.e. occupies 2 columns)

/2 = go on to card 2 (i.e. ignore the rest of card 1)

ID2 1-3 = ID2 is punched in columns 1 to 3 (don't forget: each name may appear only once, therefore ID1 and ID2, not ID twice!

V1 TO V20 5-24 = V1 is in column 5, V2 in column 6, V3 in column 7 etc. up to V20 is in column 24.

On the DATA LIST card the TO convention only works if all the variables referenced are the same width (i.e. need the same number of columns). If, for some reason, V7 were punched in two columns and all the other variables only in one, we would have to specify:

```
1                16
DATA LIST        ...
                 /2 ... V1 TO V6 5 - 10 V7 11-12 V8 TO V20 13 - 25
```

Note that on the DATA LIST card we use TO to refer to variables and "-" to refer to columns.

C. INPUT MEDIUM: it is possible to use more exotic devices such as disc or magnetic tape, consequently the input medium has to be specified.

D. N OF CASES: Obviously you change the figure given here to suit your own number of cases. On the CYBER machines you can specify:

```
1                16
N OF CASES       UNKNOWN
```

and, provided you place an EOR (End of Record) card after the last data card, the computer will count the cases for you. Ask in your computing centre whether this is possible with your machine!

E. MISSING VALUES: if there are no data missing in any of your cases, then this card is not needed. However it is generally the case that some data are missing somewhere. To designate missing data one usually chooses a number which is not otherwise required . If you simply left the column blank, the computer would read this as 0 and use this 0 as a "real" number in computations. For instance, if AGE were missing for 4 of our 12 cases, and all the remaining students were 18, the mean age of the group would be reported as:

$$\frac{(8 \times 18) + (4 \times 0)}{12} = 12$$

which would, of course, be incorrect. If however 99 has been punched and declared as a missing value, the computer would report a mean age of 18.0 for 8 cases with 4 cases missing. In the data of exhibit 1 we would run into difficulties if a student in his 9th Semester were one of the test subjects, since he would be regarded as "missing" (Value = 9) even though he is a "genuine" 9. In a case like this it would be better to make SEM a two column variable and to assign 99 as missing.

Up to 3 numbers can be assigned for missing values; this enables you to distinguish between different reasons for this value being unavailable. For example the numbers 2, 3 and 9 could be used in this way:

```
1              16
MISSING VALUES V1 TO V20 (2 3 9)
```

2 = this student arrived late and didn't take the test at all

3 = this item was ignored although the student attempted the items preceding and following it

9 = this item was not attempted because the student did not finish the test

6. Comfort for the user: labels

All elements described here are optional; however it is extremely useful to be able to recognise on the output (the printout, the list: what you get back after your data have been through the computer, with your results on it) the variables and values which you originally entered into the computer.

A. VAR LABELS

For each variable in the data a label can be included in the program which will be printed every time the variable is mentioned. These labels can have up to 40 characters (blanks and commas count as characters!) but no special characters may be used.

EXAMPLE:

1	16	80
VAR LABELS	SEX SEX OF RESPONDENT / AGE AGE OF RESPONDENT / SEM S	
	EMESTERZAHL	

As you can see from the example, while the name of the variable (here SEX, AGE, SEM) should not be split between two cards, the label can and should. For the variable SEM the label is SEMESTERZAHL. (Naturally, 80 characters do not in fact fit across the page here, but please imagine for a moment that they do!) At the end of the first card only S fitted. Only commands may begin in column 1 (the command here is VAR LABELS), consequently the continuation card begins again in column 16 with the rest of the word. If, instead of beginning at column 16 we had begun at column 20, the label on the output would read:

S EMESTERZAHL

In other words, any blanks you leave in labels are simply counted as part of the label and printed as they are arranged on the VAR LABELS card.*

* This is easy to explain. SPSS itself is actually written in FORTRAN. The internal order reads something like this: read the name of a variable. The next 40 characters - whatever they are - or all the characters until you find the next slash (/) are part of a label. Print them whenever the variable is named.

Since anything found on the VAR LABELS card after the name of a variable is considered to be part of a label and not, in fact, read or used by SPSS but simply stored, the TO convention cannot be used on the VAR LABELS card. If you do use it, you will end up with a label for the first variable which begins something like this:
ITEM1 *TO ITEM20 TEST ITEMS*
and with no labels at all for the other 19 items.

The variables need not be listed on the VAR LABELS card in the same order as they were listed on the DATA LIST card.

B. VALUE LABELS

While the VAR LABELS card is useful but not essential, since, if you are sensible, you will choose your variable names so that they have mnemonic qualities, the VALUE LABELS card is almost always indispensible because it makes the output understandable: for other people and, not least important, for you if you need to look at it again after an interval of time has passed.

By using this card we can assign a different label for every individual value of a variable. The label can be up to 20 characters long, and again, it is printed every time this value of the variable in question is mentioned on the output.
EXAMPLE:

```
1               16
VALUE LABELS    SEX (1) MALE (2) FEMALE / V1 TO V20 (1) CORRECT (2) IN
                CORRECT
```

First comes the name of the variable, then a value is specified in parentheses followed by the label for this value, then comes the next value, again in parentheses, its label etc. When all the values of one variable (or several: as the example shows the TO convention can be used on this card) have been labelled, the slash (/) is used to tell the computer to go on to the next variable.

Again the variables need not be in the same order as they were on the DATA LIST card.

If you have used various numbers to designate different reasons for values being missing, then labels for these numbers can be specified here too. Otherwise there is no need to bother, since the computer automatically labels values mentioned on the MISSING VALUES card as "MISSING".

The VALUE LABELS facility can also be used to include the names of the people who took the test in the program. EXAMPLE:*

1 16

VALUE LABELS	ID	*card 1*
	(001) JOHN	*card 2*
	(002) FRED	*card 3*
	(003) MARY (004) JOAN (005) ABBA	*card 4*
	(006) PAUL etc	*card 5*
	/ SEX (1) MALE (2) FEMALE	*card 6*
	/ etc.	*card 7*

This is useful for two reasons:

a) if you are human you will make occasional typing errors in the data. It is easier to check back if you have both name and ID number available. Also it is useful to have <u>all</u> the data included in the computer output.

b) you will be able to produce lists of test scores with the names attached - useful for posting on bulletin boards, keeping records etc.

<u>DECK PLACEMENT</u>

If present, the VAR LABELS card followed by the VALUE LABELS card is placed directly <u>before</u> the MISSING VALUES card.

---

* This set of cards shows the extreme flexibility of programming with SPSS. There is no reason, for example, why the variable ID should stand on its own on card 1 (except that in practice I have found it useful to put it there so that I don't forget to mention the name of the variable). The three values named on card 4 are just as acceptable to the computer as cards 2 and 3 where only one value is recorded. I like to ensure that the slashes are the first character on new cards so as to be able to check that I haven't forgotten them. We have plenty of cards, so don't try and be too economical!

7. Let the computer do the work: data transformation

Look at exhibit 4. Here, each of our 12 students has taken 3 tests, each test made up of 20 items. Test 1 was grammar, test 2 vocabulary, test 3 listening comprehension. Obviously it would be useful to calculate the scores for each subtest and the grand total over all three tests. Of course you can do it by hand and punch the scores in when you are preparing the punch cards, but if the computer has the results for individual items it can be told to calculate any scores wanted by using COMPUTE and COUNT cards.

A. COMPUTE cards

These use arithmetical expressions (see above, p. 8) so that if we wanted, for instance, to calculate the Intelligence Quotient on the basis of

$$IQ = \frac{\text{mental age}}{\text{chronological age}} \times 100$$

we can write:

```
1               16
COMPUTE         IQ = (MA/CA) * 100
```

The variables mentioned on the right of the = sign must be variables already known to the computer (i.e. mentioned on the DATA LIST card or created by means of COUNT or COMPUTE cards already). In this case MA and CA would have to be known variables. The variable to the left of the = sign (here: IQ) is a new variable and is created at this point by the COMPUTE card.

To compute the score for grammar in exhibit 4 we could write:

```
1               16
COMPUTE         GRA = V1 + V2 + V3 + V4 + V5 ..... + V20
```

For each case the score for GRA would be computed and added to that case after the variables entered by means of the DATA LIST.

Obviously if a test has 50 items, this card is rather

laborious to prepare. However if, for any reason, individual answers carry different numbers of points, this is the only way that the test or subtest score can be computed.

B. COUNT cards

If, on the other hand, we have a test where each item is awarded 1 point if it is correct (0 points if it is wrong) we can write:

```
1                16
COUNT            GRA = V1 TO V20 (1)
```

This tells the computer that every time it discovers a "1" it is to add 1 to the score for GRA. It is important to realise that for the COUNT procedure the "1" is merely a symbol, not a number, in other words for COUNT you could equally well use something like "§" and tell the computer to count the number of times this occurs. Therefore, if a correct answer, for instance, is scored 2, an incorrect answer 0, and there are no other scores (if there are, use COMPUTE) we could compute the score for the test by writing:

```
1                16
COUNT            TEMP = V1 TO V20 (2)
COMPUTE          GRA = TEMP * 2
```

In this case the computer adds 1 to TEMP (a temporary variable) every time it discovers a 2. In other words, if it finds 12 "2"s, TEMP will have a value of 12. To obtain the final score this must be multiplied by 2 to give GRA = 24.

COUNT and COMPUTE cards can be used in any order they are needed, and a new variable created by one card is immediately added to the end of each case and can be used in the next COUNT or COMPUTE command.

In order to calculate the subtest and total scores for exhibit 4 we would write:

```
1                16
COUNT            GRA = V1 TO V20 (1)
COUNT            VOC = V21 TO V40 (1)
COUNT            LIST = V41 TO V60 (1)
COMPUTE          TOTAL = GRA + VOC + LIST
```

Since these variables are now attached to the cases, the command:

```
1                16
FREQUENCIES      GENERAL = GRA TO TOTAL
```

would cause the computer to perform the procedure FREQUENCIES for all four new variables.

MISSING DATA in data transformations

There is, unfortunately, a problem when missing data are involved in a data transformation. If the computer discovers that any one of the variables involved, for example, in a COUNT list has missing data, it automatically sets the value of the new variable to 0. It is as though the computer says, "I cannot calculate the value of GRA for this case because V7 = 9, 9 is being used for missing data, therefore the data are not complete. Consequently I will assign 0 to the case. What it then forgets to do (at least this is an SPSS weakness, the computer is not at fault as such) is to note down somewhere that it has given the case a score of 0 because data were missing. So that the next time the variable is used all scores of 0 are treated as though the 0 were a real number, and consequently they enter into all calculations (see example above, p. 12). To stop this happening we must, at the end of a transformation, specify a new value which we wish to be regarded as missing. 0 is a bad choice because it is conceivable that someone really did get 0 items right, and has a genuine score of 0. So once again we have recourse to a score which is logically impossible:

```
1               16
ASSIGN MISSING GRA TO TOTAL (99)
```

This card follows immediately after the data transformation cards and puts matters right again.

This block of cards: COUNT, COMPUTE and ASSIGN MISSING are placed immediately after the N OF CASES card, which is logical because you may well want to assign labels to the newly-created variables as well as to those you originally put in with the DATA LIST card.

8. Entering data

On the majority of computers SPSS expects, after the data definition and preliminary data manipulation block has been completed, to receive the first command for a statistical operation. Only after it has been told to do something with the data will it condescend to read the data. On the CYBER 76 it is possible to read the data first and then to call for the first statistical procedure to be carried out. However even on the CYBER it is possible to insert the first procedure card (control card for a statistical operation) before reading in the data.

Consequently the layout of the program (cf. page 2 above) is:

```
DATA DEFINITION AND PRELIMINARY MANIPU-
                    LATION
```

```
Procedure 1, Options, Statistics
```

followed by the data block. This consists quite simply of the control card:

```
1              16
READ INPUT DATA
```

followed by your data cards.

NOTE

If your computer has the N OF CASES UNKNOWN facility (which means that the computer will count the number of cases for you) you will need an EOR (End of Record) card after the last data case, otherwise the computer will try and read the procedure cards which follow the data as data cards, will discover alphabetic characters where it expects to find numeric ones and will break off the run.

Summary of cards discussed so far

The cards necessary for the data definition and preliminary data manipulation should appear in the program in the following order:

	CARD	STATUS
1.	RUN NAME	obligatory
2.	DATA LIST	obligatory
3.	INPUT MEDIUM	obligatory
4.	N OF CASES	obligatory
5.	*data transformation cards*	optional
	COUNT	
	COMPUTE	
	other data transformation cards: see McGraw-Hill Manual	
	ASSIGN MISSING	
6.	VAR LABELS	optional
7.	VALUE LABELS	optional
8.	MISSING VALUES	optional
9.	*PROCEDURE 1*	obligatory*
	OPTIONS	(optional)
	STATISTICS	(optional)
10.	READ INPUT DATA	obligatory
	data data data data data	obligatory
11.	*PROCEDURE 2, 3, 4 etc.*	optional
	OPTIONS	
	STATISTICS	
12.	FINISH	obligatory

* Obviously at least <u>one</u> statistical procedure is obligatory, otherwise there would be no need to run the cards through the computer.

9. Statistical procedures in SPSS

All statistical procedures in SPSS are called by using a procedure card which has as its command the name of the procedure involved. Each procedure card can be followed by an options card and a statistics card, but need not be. The individual procedures can be used in any order in the program. In the specification field of the procedure card the names of the variables on which the procedure is to be carried out are listed.

SPSS is relatively inflexible in that the procedure cards have to be set up exactly as specified in the Manual (or here). Also data resulting from one calculation can not be immediately entered into another calculation. Hoever the great advantage of SPSS is that one need not even know the formula and the steps in calculation to use extremely complex procedures for data analysis such as factor analysis or multiple regression. This is, unfortunately, also its great disadvantage: it is very easy to misuse these procedures if their assumptions are not met. So a word of warning may be timely: by all means play with SPSS, but don't use the more complicated procedures for real research unless you have studied them in some detail.

Almost all the relevant test statistics can be produced by putting together various statistical procedures from the SPSS repertoire. For those which cannot be produced from SPSS programs, such as the standard error of measurement or a detailed analysis of multiple-choice items, a good (I would recommend programmable) pocket calculator is an excellent investment, since even in those cases where SPSS does not produce the final figure it takes the analysis quite a long way.

The following is a list of the main test statistics needed for basic test analysis, and of the SPSS procedures involved in producing these statistics. Finally each procedure is described in sufficient detail to enable you to use it in your own programming. All procedures will be demonstrated using the output of exhibit 4.

TABLE OF TEST STATISTICS AND SPSS PROCEDURES

test statistic		SPSS procedure
mean, mode, median	of tests/subtests	FREQUENCIES
variance	"	
standard deviation	"	
numbers of correct, incorrect and missing responses to individual items		FREQUENCIES
item difficulties		RELIABILITY
item discrimination indices		
reliability coefficients (KR 20)		
reliability coefficients (split-half)		
intercorrelations between subtests, between subtests and totals		PEARSON CORR
test-retest reliability (by correlation)		
criterion/test correlation (validity)		
any other correlative measures		
analysis of the relationships between demographic variables and test results		BREAKDOWN T-TEST
analysis of demographic variables		FREQUENCIES
lists of test results		BREAKDOWN LIST CASES
error routines		EDIT LIST CASES BREAKDOWN

Full details of all these procedures are given in the McGraw-Hill Manual with the exception of RELIABILITY. This is described in Beutel.

COMPLETE PROGRAM FOR EXHIBIT 4
for comparison with computer listing

```
1              16
RUN NAME       EXHIBIT 4 TEST ANALYSIS
DATA LIST      FIXED (1) /1 ID 1-3 AGE 4-5 SEX 6 NOTE 7 V1 TO V60 8 - 67
INPUT MEDIUM   CARD
N OF CASES     13 (One case where all data are missing was added)
COUNT          GRA = V1 TO V20 (1)
COUNT          VOC = V21 TO V40 (1)
COUNT          LIST = V41 TO V60 (1)
COMPUTE        TOTAL = GRA + VOC + LIST
ASSIGN MISSING GRA TO TOTAL (99)
VAR LABELS     NOTE ABITURNOTE/ GRA GRAMMAR TEST / VOC VOCABULARY TEST
               /LIST LISTENING COMPREHENSION
               /TOTAL TOTAL TEST SCORE
VALUE LABELS   SEX (1) MALE (2) FEMALE / NOTE (1) SEHR GUT (4) AUSREICHEND
               /ID (001) JOHN (002) FRED (003) MARY (004) JOAN (005) ABBA
               (006) PAUL (007) JANE (008) JUDE (009) PETE (010) TONY
               (011) KRIS (012) DOUG
MISSING VALUES AGE (99) / SEX TO V60 (9)
* FREQUENCIES  GENERAL = AGE TO NOTE GRA TO TOTAL
  OPTIONS      8
  STATISTICS   ALL
  READ INPUT DATA
  data data data data data data data data data data data data data (13 cards!)
* RELIABILITY  VARIABLES = V1 TO V60/ SCALE (GRA) = V1 TO V20 /SCALE (VOC)
               = V21 TO V40/ SCALE (LIST) = V41 TO V60
               /MODEL = ALPHA
  OPTIONS      15
  STATISTICS   1 4 TO 17
* PEARSON CORR GRA TO TOTAL
  BREAKDOWN    TABLES = GRA TO TOTAL BY AGE NOTE
  STATISTICS   ALL
* T-TEST       GROUPS = SEX (1 2) / VARIABLES = GRA TO TOTAL
* BREAKDOWN    TABLES = GRA TO TOTAL BY ID
  FINISH
```

* Procedures used: FREQUENCIES, RELIABILITY, PEARSON CORR, BREAKDOWN, T-TEST, BREAKDOWN

10. FREQUENCIES

FREQUENCIES counts the number of objects which fall into different categories. For each variable it reports the code (accompanied by the label, if there is one), the absolute frequency of objects in each category, the relative frequency as a percentage of the total, the adjusted frquency (i.e. all valid cases are counted as 100%, missing data are ignored) and the cumulative frequency. The control card is:

```
1               16
FREQUENCIES     GENERAL = AGE SEX NOTE GRA TO TOTAL
OPTIONS         8
STATISTICS      ALL
```

The results of this command can be seen in the output (next page). For each of the variables the above-mentioned statistics have been produced. Option 8 makes the machine produce a histogram so that we have an optical picture of the data. Calling for all statistics produces: MEAN, STANDARD ERROR, MEDIAN, MODE, STANDARD DEVIATION, VARIANCE, KURTOSIS, SKEWNESS, RANGE, MINIMUM, MAXIMUM and the 95% CONFIDENCE INTERVAL around the mean.

With most statistical procedures it is safe to call for

```
STATISTICS    ALL
```

however be careful not to ask for

```
OPTIONS       ALL
```

because in many cases some options are mutually exclusive, and the computer would break off the job.

Any test analysis will involve several runs through the computer, and one of the errors which FREQUENCIES can detect is mispunched data. In your first run, therefore, you should include the command:

```
1               16
FREQUENCIES     GENERAL = ALL
```

(No options or statistics as this would waste time and space).

==========output from program FREQUENCIES==========

EXHIBIT 4 TEST ANALYSIS

FILE NONAME (CREATION DATE = 02/03/79)

NOTE ABITURNOTE (name of variable followed by the label we entered)

CATEGORY LABEL	CODE	ABSOLUTE FREQ	RELATIVE FREQ (PCT)	ADJUSTED FREQ (PCT)	CUM FREQ (PCT)
SEHR GUT (we entered this label)	1.	2	15.4	16.7	16.7
	2.	2	15.4	16.7	33.3
	3.	4	30.8	33.3	66.7
AUSREICHEND	4.	4	30.8	33.3	100.0
	9.	1	7.7		
	TOTAL	13	100.0	100.0	

- - - -

If, for instance, under SEX you discover the following:

CATEGORY LABEL	CODE	ABSOLUTE FREQ
MALE	1	24
FEMALE	2	29
	4	1
MISSING	9	3

the 4 is probably a punching error and this must be eliminated before further analysis is carried out. Use the error routine (see below, p. 34) to discover which card has been mispunched.

TO SUM UP:

To make the computer perform the operation of counting you give it the command

```
1              16
‡ FREQUENCIES    GENERAL = the variables you want to be counted
```

11. RELIABILITY

This procedure reports:

- item difficulties (p values)
- item discrimination indices (part/residual correlations)
- reliability according to the model specified:
  - a) Kuder-Richardson 20 in Cronbach's extension ALPHA
  - b) Split-half reliability according to the odd-even method and also the corrected reliability after use of the Spearman-Brown Correction Formula
- standard deviation of the test
- variance of the test
- mean score obtained on the test
- mean difficulty of the items
- mean inter-item correlation

and various other statistics which are useful if you know how to interpret them.

The figures reported in the RELIABILITY output may differ from those reported by FREQUENCIES because all items with zero variance (i.e. which everyone got right or wrong)

‡ There are very good reasons why GENERAL has to be the first word in the specification field here. If you are interested, read the McGraw-Hill Manual!

are removed before the analysis begins.

The control card for RELIABILITY consists of 3 pieces of information the computer needs to know before it can start work:

(i) which variables are involved in the analysis

(ii) how are they to be combined into tests (scales) and what are these scales to be called

(iii) what type of reliability analysis do you want the computer to perform

Consequently the control card looks like this:

1	16	
RELIABILITY	VARIABLES = V1 TO V20	*answer to question (i)*
	/SCALE (GRA) = V1 TO V20	*answer to question (ii)*
	/MODEL = ALPHA	*answer to question (iii)*
OPTIONS	15	
STATISTICS	1 4 TO 17	

This is the point at which the inflexibility of SPSS seems to be most apparent; however there are excellent reasons for constructing the control card in this way. In the first place up to 100 variables (on most computers actually more than 100) can be mentioned in the VARIABLES = part of the card. This means that in the SCALE = part more than one scale can be defined, thus reducing control card preparation. In the MODEL = part of the card the model required must be specified unless the model is ALPHA, which the computer always assumes you want unless you specify otherwise.

The RELIABILITY procedure enables you to experiment with your data in order to construct the best tests possible out of the items at your disposal. Suppose, for instance, you had administered 20 items and you wanted to construct the most reliable 10 item test out of them. The original output for the 20 items would be the basis for the first sorting, and then you could play with RELIABILITY until you found the best set of items:

RELIABILITY	VARIABLES = V1 TO V20/SCALE (TRY1) = V1 V4 V7.../SCALE
	(TRY2) = V3 V7 V12 V14 TO V20/SCALE (TRY3) = V1 V7...

information about items produced by RELIABILITY

		MEANS		STD DEV	CASES
1.	V21	.83333		.38925	12.0
2.	V22	.91667		.28868	12.0
3.	V23	.91667		.28868	12.0
4.	V24	.58333		.51493	12.0
5.	V25	.75000		.45227	12.0
6.	V26	.75000	item	.45227	12.0
7.	V27	.91667	difficulties	.28868	12.0
8.	V28	.83333		.38925	12.0
9.	V29	.66667		.49237	12.0
10.	V30	.91667		.28868	12.0
11.	V31	.66667		.49237	12.0
12.	V32	.50000		.52223	12.0
13.	V33	.83333		.38925	12.0
14.	V34	.33333		.49237	12.0
15.	V35	.33333		.49237	12.0
16.	V36	.58333		.51493	12.0
17.	V37	.58333		.51493	12.0
18.	V38	.33333		.49237	12.0
19.	V39	.25000		.45227	12.0
20.	V40	.33333		.49237	12.0

	SCALE MEAN IF ITEM DELETED	SCALE VARIANCE IF ITEM DELETED	CORRECTED ITEM-TOTAL CORRELATION	ALPHA IF ITEM DELETED
			discrimination	index
V21	12.00000	23.45455	.57869	.88577
V22	11.91667	24.26515	.50611	.88847
V23	11.91667	24.99242	.24672	.89333
V24	12.25000	22.20455	.68376	.88161
V25	12.08333	22.26515	.77743	.87921
V26	12.08333	22.62879	.68665	.88204
V27	11.91667	24.62879	.37545	.89094
V28	12.00000	23.09091	.68044	.88309
V29	12.16667	21.78788	.81749	.87718
V30	11.91667	24.99242	.24672	.89333
V31	12.16667	23.78788	.36595	.89218
V32	12.33333	22.42424	.62493	.88367
V33	12.00000	24.90909	.18718	.89568
V34	12.50000	24.81818	.14825	.89896
V35	12.50000	22.45455	.66239	.88248
V36	12.25000	22.56818	.60390	.88441
V37	12.25000	22.02273	.72419	.88017
V38	12.50000	23.72727	.37905	.89176
V39	12.58333	24.08333	.33792	.89252
V40	12.50000	23.00000	.53899	.88658

EXHIBIT 4 TEST ANALYSIS

FILE NONAME (CREATION DATE = 02/03/79)

**********RELIABILITY ANALYSIS FOR SCALE (VOC

N OF CASES = 12.0

STATISTICS FOR	MEAN	VARIANCE	STD DEV	N OF VARIABLES
SCALE	12.83333	25.78788	5.07818	20

ITEM MEANS	MEAN	MINIMUM	MAXIMUM	RANGE
	.64167	.25000	.91667	.66667

ITEM VARIANCES	MEAN	MINIMUM	MAXIMUM	RANGE
	.19621	.08333	.27273	.18939

INTER-ITEM COVARIANCES	MEAN	MINIMUM	MAXIMUM	RANGE
	.05754	-.12121	.22727	.34848

INTER-ITEM CORRELATIONS	MEAN	MINIMUM	MAXIMUM	RANGE
	.28698	-.50000	1.00000	1.50000

RELIABILITY COEFFICIENTS 20 ITEMS

ALPHA = .89245 STANDARDIZED ITEM ALPHA = .88950

information about the test produced by RELIABILITY 30

12. PEARSON CORR

This procedure produces Pearson product-moment correlation coefficients for all the variables listed on the card. Consequently the control card:

```
1              16
PEARSON CORR    GRA TO TOTAL
```

will produce the following output:

	GRA	VOC	LIST	TOTAL
GRA	1.0000	.9305	.9642	.9833
	(0)	(12)	(12)	(12)
	S= .001	S= .001	S= .001	S= .001
VOC	.9305	1.0000	.9170	.9739
	(12)	(0)	(12)	(12)
	S= .001	S= .001	S= .001	S= .001
LIST	.9642	.9170	1.0000	.9789
	(12)	(12)	(0)	(12)
	S= .001	S= .001	S= .001	S= .001
TOTAL	.9833	.9739	.9789	1.0000
	(12)	(12)	(12)	(0)
	S= .001	S= .001	S= .001	S= .001

(COEFFICIENT / CASES / SIGNIFICANCE)

(VOC and LIST are the two variables between GRA and TOTAL)

There are no statistics for PEARSON CORR.

13. BREAKDOWN

The breakdown procedure enables us to place to variables in relationship to each other. For instance we might be interested in the relationship between test results and demographic variables. Do people with high marks in the ABITUR also have high test scores? BREAKDOWN uses a command

output produced by BREAKDOWN

CRITERION VARIABLE LISTENING COMPREHENSION TEST
BROKEN DOWN BY NOTE ABITURNOTE .

VARIABLE	CODE	VALUE LABEL	SUM	MEAN	STD DEV	VARIANCE	N
FOR ENTIRE POPULATION			155.0000	12.9167	4.2525	18.0833	(12)
NOTE	1.	SEHR GUT	35.0000	17.5000	.7071	.5000	(2)
NOTE	2.		33.0000	16.5000	.7071	.5000	(2)
NOTE	3.		54.0000	13.5000	3.0000	9.0000	(4)
NOTE	4.	AUSREICHEND	33.0000	8.2500	2.2174	4.9167	(4)

TOTAL CASES = 13
MISSING CASES = 1 OR 7.7 PCT.

lists produced by BREAKDOWN

CRITERION VARIABLE GRA DESCRIPTIVE SUBPOPULATIONS
BROKEN DOWN BY ID GRAMMAR TEST

VARIABLE	CODE	VALUE LABEL	SUM	MEAN	STD DEV	VARIANCE	N
FOR ENTIRE POPULATION			158.0000	13.1667	3.9505	15.6061	(12)
ID	1.	JOHN	10.0000	10.0000	0.0000	0.0000	(1)
ID	2.	FRED	17.0000	17.0000	0.0000	0.0000	(1)
ID	3.	MARY	16.0000	16.0000	0.0000	0.0000	(1)
ID	4.	JOAN	16.0000	16.0000	0.0000	0.0000	(1)
ID	5.	ARBA	6.0000	6.0000	0.0000	0.0000	(1)
ID	6.	PAUL	12.0000	12.0000	0.0000	0.0000	(1)
ID	7.	JANE	18.0000	18.0000	0.0000	0.0000	(1)
ID	8.	JUDE	17.0000	17.0000	0.0000	0.0000	(1)
ID	9.	PETE	8.0000	8.0000	0.0000	0.0000	(1)
ID	10.	TONY	10.0000	10.0000	0.0000	0.0000	(1)
ID	11.	KRIS	15.0000	15.0000	0.0000	0.0000	(1)
ID	12.	DOUG	13.0000	13.0000	0.0000	0.0000	(1)

of the form:

1	16
BREAKDOWN	TABLES = GRA TO TOTAL BY NOTE AGE
STATISTICS	ALL

First each category of GRA is broken down by NOTE, then each category of GRA by AGE, then the next variable VOC is broken down by NOTE and then by AGE and so on. Naturally it is also possible to use BREAKDOWN to compare test results with other test results (e.g. BREAKDOWN of GRA by LIST etc.) or demographic variables with other demographic variables (BREAKDOWN of NOTE by AGE etc.).

The request for all statistics produces a one-way analysis of variance and a test for linearity. This is, however, not the point to go into inferential statistics, and if you don't understand this part of the program, then simply omit the STATISTICS card.

14. T-TEST

When inferential statistics are needed for two groups only the t-test is used rather than the analysis of variance. For more details about this card see the McGraw-Hill Manual.

15. BREAKDOWN for lists

By using the variable ID in a BREAKDOWN procedure we can produce lists of test scores, if you have entered the names as values in the VALUE LABELS cards even lists with names.

16. FINISH

Logically enough this is the final card in all SPSS runs.

17. EDIT

Nobody is perfect, and most people make silly mistakes the first (and 200th) time they write a new program. Typing errors occur, slashes are forgotten, numbers appear as letters and letters as numbers. You will be surprised at the time you have to spend persuading a high-speed computer to accept your program: he's a perfectionist. So the time spent in the computing centre can turn out to be much more than you originally intended to spend there. One of the ways of ensuring that you don't waste more time than is necessary is by planning the things you have to do in a logical order and letting the computer help you.

Programs are probably best written at home. However once written, a program has to be punched. You also have to punch the data. If you start by punching the program, you can have the SPSS EDIT facility check your syntax and the logical structure of the program, and while you are waiting for this to reemerge from the machine room or printer, you can be typing the data.

For an EDIT run all you need is the basic program (e.g. the one given in this paper for exhibit 4) without the READ INPUT DATA card (and of course without the data). Otherwise the program stays just as it is, except that the very first card - before the RUN NAME card - is:

```
1               16
EDIT
```

The SPSS compiler will diagnose your mistakes - though sometimes one needs help in interpreting the diagnosis - and help you to get the program run. When you finally receive an output that tells you

NUMBER OF ERRORS DETECTED 0

you can remove the EDIT card, insert the READ INPUT DATA card and the data at the right place in the card deck and send it through for the first proper run.

18. Error routines

If, in the course of the first run, where the only procedure is:

```
1               16
FREQUENCIES     GENERAL = ALL
```

you discover you have made a punching error you can discover which card and case is involved by removing all procedure cards and inserting the following mini-program before the READ INPUT DATA card:

```
1               16
*SELECT IF      (SEX EQ 4)
LIST CASES      CASES = 10/ VARIABLES = ID SEX
BREAKDOWN       SEX BY ID
```

What happens is that first the case where SEX equals (EQ) 4 is identified (naturally you insert here the variable and the value you are concerned with instead of SEX and 4). Then all the cases fitting this condition are listed by ID. We have, in fact, stated that 10 cases are to be listed, but this is a generous estimate. Then the BREAKDOWN procedure will, if the names are present in the program as value labels, tell you which person is involved, so that you can go back to the raw data (test scripts or whatever) and discover which value should have been punched. The card is corrected (duplicated and the correct value inserted at the appropriate position) and the next run can start.
You may not omit the BREAKDOWN procedure, even if it seems senseless, because LIST CASES will not run on its own.
If you discover more than one mispunched card and different variables are affected, you can put in more than one error detection block in a run: each block counts as one procedure.

WRITE CASES

By using this facility it is possible to make the computer print the data as entered so that you can check them. Ask the SPSS adviser at your computing centre how this works on your machine.

19. A final word

Computing is fun if you don't let yourself get upset about making mistakes now and then. You will find the people in your computing centre are only too willing to help you if you have problems - and everyone has problems at the beginning. However if you intend to go into testing seriously, driving a computer can take most of the drudgery out of testing. Naturally enough, most language specialists would rather learn to use a new language to talk to a machine than to spend hours and hours with pencil and paper doing sums.

If you now want to know more about SPSS read the McGraw-Hill Manual. My aim has been to introduce you to the basic minimum necessary for test analysis by computer, but also to show you that computing can be a lot easier than you think it is.

Have fun!

EXHIBIT 1

NAME	AGE	SEX	SEMESTER ZAHL	QUESTION NUMBER IN TEST 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
JOHN	19	male	3	1	1	1	1	1	1	0	0	0	0	-	-	-	-	-	-	-	-	-	-
FRED	20	male	4	1	1	1	1	1	0	1	1	1	1	0	0	1	1	1	1	1	0	0	0
MARY	18	female	3	1	1	1	1	1	1	0	0	1	1	1	1	1	-	0	0	0	0	0	0
JOAN	22	female	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	-
ABBA	19	female	2	1	1	1	1	0	0	0	-	-	-	0	0	1	1	1	0	-	-	-	-
PAUL	20	male	3	1	1	1	0	0	1	1	1	0	0	1	1	1	0	0	0	0	0	0	0
JANE	24	female	7	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	-	1	1	1	1
JUDE	19	female	1	1	1	1	1	1	0	0	0	-	-	-	0	0	0	1	1	0	0	0	0
PETE	22	male	4	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	0	0
TONY	24	male	5	1	0	0	1	1	1	0	1	1	1	1	0	0	0	0	1	1	1	0	0
KRIS	21	female	6	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	1	1	0
DOUG	27	male	8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1

37

EXHIBIT 2

Coding Plan

Card	Variable(s)	Columns	Coding
1	ID1	1 - 4	0001 to N
	CN1	5	1
	JAHR (des Abitur)	6 - 9	as given; missing = 9999
	QUAL (Studienberechtigung)	10	1 = Abitur 9 = missing 2 = Fachhochschulreife 3 = Begabtensonderprüfung 4 = Sonstiges
	ZIEL (Angestrebter Studienabschluß)	11	1 = Sekundarstufe 1 9 = missing 2 = Sekundarstufe 2 3 = Magister 4 = Promotion 5 = Sonstiges
	STAY (Aufenthalt in engl.-sprech. Land)	12	1 = yes 9 = missing 0 = no
	SEX	13	1 = male 9 = missing 2 = female
2	ID2	1 - 4	0001 to N
	CN2	5	2
	ITEM1 to ITEM20 (Vocab. test)	6 - 25	1 = correct 9 = missing 0 = incorrect
	ITEM21 to ITEM40 (Grammar test)	26 - 45	as above

227 people took the two tests

38

EXHIBIT 3

A new course COMPREHENSION is to be introduced which will be obligatory for all students. To discover whether, in fact, this course does produce an increase in the skill of "reading comprehension" it is decided to use a pretest - posttest design. Both the experimental and the control group will be administered a 30 item cloze test and a 20 item multiple-choice reading comprehension test at the beginning and at the end of the semester (course). The experimental group (100) will receive teaching in the new course. The control group (50) will not receive teaching in COMPREHENSION. They will, however, be attending other courses in the English department. In addition to their names and test scores the students will be expected to provide: number of hours of study in the English department in the semester concerned; number of Semesters English already studied; sex.

EXHIBIT 4

ID	AGE	SEX	NOTE	Grammar 20 items	Vocabulary 20 items	Listening Comprehension 20 items
001	18	1	4	11101111100100010000	11111011010010001000	11101111010110001000
002	20	2	1	11111111111110111100	11111111111110011111	11111111111011111001
003	19	2	2	11111111111111001100	11111111111110111100	11111111111100111010
004	18	2	3	11101101111100111000	11111111001100110100	10011100111100111010
005	19	2	4	10010010001110000000	11100000011010000000	11000011100000010000
006	20	1	3	11111110110011100000	11111111111000000100	11111111110001000100
007	19	1	2	11111111111111011011	11111111111111111011	11111111111111111000
008	20	1	1	11111111111011110110	11111111111111111001	11111111111110011111
009	19	2	4	01111100010001000100	01000111001101010000	00100010111001001000
010	18	1	4	00100010011111111000	00100010010010000000	00011110110101000100
011	19	2	3	11111111110011101010	11101111110011001001	01111111101111110101
012	20	2	3	11111111010001110100	11101111110010010010	01000111001011110100

1 - 3 4-5 6 7 8 - 27 28 - 47 48 - 67

Grammar Test = GRA; V1 to V20; punched in cols 8 - 27
Vocabulary Test = VOC; V21 to V40; punched in cols 28 - 47
Listening Comprehension Test = LIST; V41 to V60; punched in cols 48 - 67
Demographic Variables: Age; Sex; Abiturnote

EXERCISE 1

WORK OUT THE CODING FOR EXHIBIT 3

STEP 1 DECIDE WHICH VARIABLES HAVE TO BE INCLUDED IN THE PROGRAM

STEP 2 DECIDE WHICH SPSS NAMES YOU ARE GOING TO GIVE THE VARIABLES

STEP 3 DECIDE HOW MANY COLUMNS EACH VARIABLE NEEDS

STEP 4 DECIDE WHAT VALUES CAN BE OBTAINED AND HOW YOU ARE GOING TO CODE THESE

STEP 5 DECIDE WHICH NUMBERS YOU ARE GOING TO USE FOR MISSING VALUES

STEP 6 DECIDE WHAT YOU ARE GOING TO PUT ON WHICH CARD

CARD	VARIABLE	COLUMNS	CODING

LOOK AT EXHIBIT 2 IF YOU ARE HAVING DIFFICULTIES

EXERCISE 2

PREPARE THE DATA DEFINITION PARTS OF THE PROGRAMS FOR EXHIBITS 2 AND 3

EXHIBIT 2

1 16
RUN NAME
DATA LIST
INPUT MEDIUM CARD
N OF CASES
MISSING VALUES

EXHIBIT 3

1 16

EXERCISE 3

PREPARE SUITABLE VAR LABELS AND VALUE LABELS FOR EXHIBITS 2 AND 3

N.B. USE LABELS ONLY WHEN THEY ARE GENUINELY USEFUL; I.E. THERE IS VERY LITTLE POINT IN INFORMING YOURSELF (OR ANYONE ELSE WHO READS THE OUTPUT) THAT AGE = AGE. BUT YOU MIGHT WANT TO KNOW THAT ZUFZ = ZUFALLSZAHL.

EXHIBIT 2

1 16

VAR LABELS

VALUE LABELS

Exhibit 3

1 16

EXERCISE 4

PREPARE THE CARDS NECESSARY FOR PRODUCING THE SUBTEST TOTALS AND GRAND TOTALS FOR EXHIBITS 2 AND 3

EXHIBIT 2

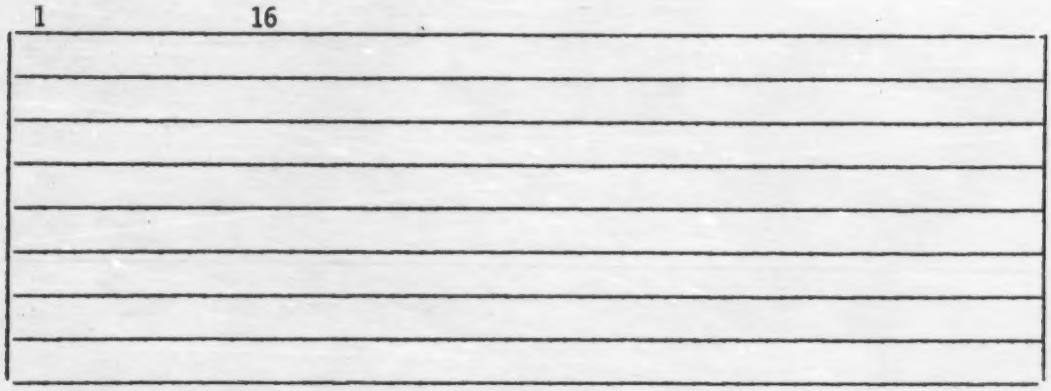
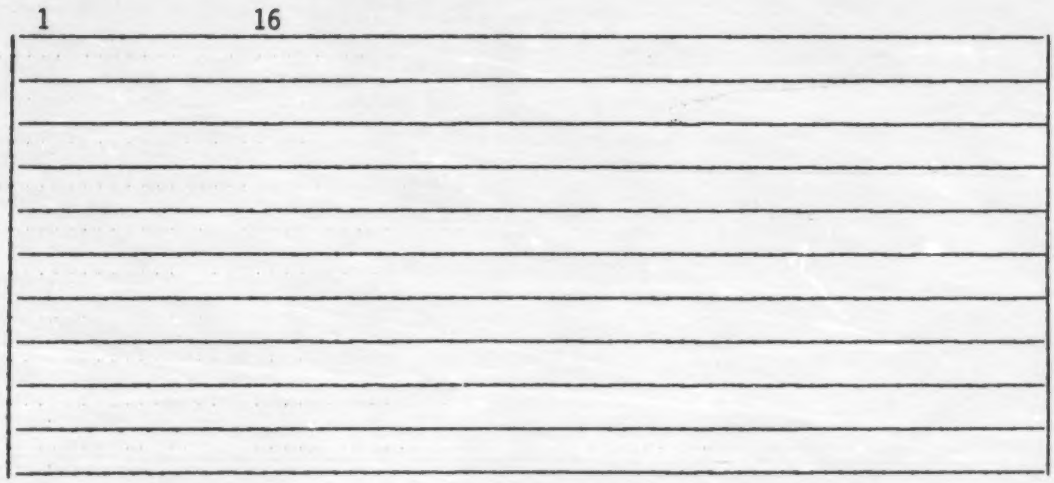


EXHIBIT 3



EXERCISE 1 SOLUTION

See also complete solution to exhibit 3, p. 49 below.

CARD	VARIABLE	COLUMNS	CODING	MISSING
1	ID1	1 - 4	0001 - 0150	
	CN1	5	1	
	SEX	6	1 = männl. 2 = weibl.	9
	SEM (Semester)	7 - 8	as given	99
	SWOCHST (Sem.wochenstd.)	9 -10	as given	99
	GROUP	11	1 = Exp. 2 = Kontr.	?(9)
2 (erste Testdurchführung am Anfang des Semesters)				
	ID2	1 - 4	as above, card 1	
	CN2	5		
	CL11 TO CL130*	6 - 35	1 = right, 0 = wrong	9
	MC11 TO MC120*	36 - 45	1 = right, 0 = wrong	9
3 (2nd test run at the end of the Semester)				
	ID3	1 - 4		
	CN3	5		
	CL21 TO CL230*	6 - 35		
	MC21 TO MC220*	36 - 45		

* See explanation in complete solution, p. 49 below.

EXERCISE 2 SOLUTION

1	16
RUN NAME	ANALYSIS OF TWO TESTS (EXHIBIT 2)
DATA LIST	FIXED (2)
	/1 ID1 1 - 4 CN1 5 JAHR 6 - 9 QUAL 10 ZIEL 11
	STAY 12 SEX 13
	/2 ID2 1 - 4 CN2 5 ITEM1 TO ITEM20 6 - 25
	ITEM21 TO ITEM40 26 - 45
N OF CASES	227
MISSING VALUES	JAHR (9999)/QUAL (9)/ZIEL (9)/STAY (9)/SEX (9)
	/ITEM1 TO ITEM40 (9)

=◊=

For exhibit 3:

SEE COMPLETE SOLUTION, p. 49 below.

EXERCISE 3 SOLUTION

For exhibit 2:

1	16
VAR LABELS	JAHR JAHR DES ABITURS,/ QUAL STUDIENBERECHTIGUNG /
	ZIEL ANGESTREBTER STUDIENABSCHLUSS/ STAY AUFENTHALT IN
	EINEM ENGLISCHSPRECHENDEN LAND
VALUE LABELS	QUAL (1) ABITUR (2) FACHHOCHSCHULREIFE (3) BEGABTENSOND
	ERPRUEFUNG (4) SONSTIGES / ZIEL (1) SEK 1 (2) SEK 2
	(3) MAGISTER (4) PROMOTION (5) SONSTIGES /STAY (1) JA
	(0) NEIN/ SEX (1) MAENNL. (2) WEIBL. /
	ITEM1 TO ITEM40 (1) RICHTIG (0) FALSCH

For exhibit 3:

SEE COMPLETE SOLUTION, p. 49 below

EXERCISE 4 SOLUTION

For exhibit 2

1	16
COUNT	VOCAB = ITEM1 TO ITEM20 (1)
COUNT	GRAM = ITEM21 TO ITEM40 (1)
COMPUTE	TOTAL = VOCAB + GRAM
ASSIGN MISSING VOCAB GRAM TOTAL (9 8)	

For exhibit 3:

SEE COMPLETE SOLUTION, p.49 below.

Exhibit 3 Solution

The variables you need to collect are:

NAME (ID)	3 or 4 columns, I always use 4 (then cards are always compatible)
SEX	1 = male, 2 = female, 9 = miss.
SEMESTERZAHL (SEM)	2 columns, 99 = missing
SEMESTERWOCHENSTUNDEN (SWOCHST)	2 columns, 99 = missing
GROUP	1 = experimental, 2 = control (should be no missing!)
CLOZE1	CL11 TO CL130 (add with computer) = CL1, 1 col. per item
MULTCH1	MC11 TO MC120 (" ") = MC1
CLOZE2	CL21 TO CL230 (" ") = CL2
MULTCH2	MC21 TO MC220 (" ") = MC2
TOTAL1	CL1 + MC1 (add with computer)
TOTAL2	CL2 + MC2

I would use 3 cards for this, on the first I would put the demographic variables, on the 2nd the 1st test and on the 3rd the 2nd test:

1	16
RUN NAME	COMPREHENSION RESEARCH, SOMMERSEMESTER 1978
DATA LIST	FIXED (3)
	/1 ID 1-4 CN 5 SEX 6 SEM 7-8 SWOCHST 9-10 GROUP 11
	/2 ID2 1-4 CN2 5 CL11 TO CL130 6 - 35 MC11 TO MC120 36
	- 45
	/3 ID3 1 - 4 CN3 5 CL21 TO CL230 6 - 35 MC21 TO MC220
	36-45
INPUT MEDIUM	CARD
N OF CASES	150
COUNT	CL1 = CL11 TO CL130 (1)
COUNT	MC1 = MC11 TO MC120 (1)
COUNT	CL2 = CL21 TO CL230 (1)
COUNT	MC2 = MC21 TO MC220 (1)
COMPUTE	TOTAL1 = CL1 + MC1
COMPUTE	TOTAL2 = CL2 + MC2
ASSIGN MISSING	CL1 TO TOTAL2 (99)
VAR LABELS	SEM SEMESTER BEREITS STUDIERT ANGL/SWOCHST SEMESTERWOC
	HENSTUNDEN
VALUE LABELS	SEX (1) MAENL. (2) WEIBL. / GROUP (1) EXPERIMENTAL
	(2) KONTROLLGRUPPE
MISSING VALUES	SEX (9) / SEM SWOCHST (99)/ CL11 TO MC220 (9)
PROCEDURE	*anything you feel is necessary*
READ INPUT DATA	

Exhibit 3 Solution

Comments

ID (NAME) — Since the computer only reads what you tell it, you can, if you keep the demographic details all on one card, type the name of the student on the punch card. If you don't tell the computer to read it, it will ignore it.

Much of my time is spent moving cards around, forming sub-groups of students etc. and finding out how they performed on other tests. If the names are on the ID cards it is always much easier to do this sort of ex post facto research.

CLOZE1 — Don't let these names confuse you
CLOZE test 1 Item 1 becomes CL11
CLOZE test 1 Item 30 becomes CL130

in the program:

DATA LIST — If you split the DATA LIST card up like this, you can run the data for the first test through as soon as you have them. You simply leave the /3 out and change FIXED (3) to FIXED (2). (I couldn't wait a whole semester to find out what had emerged from the pretest!)

Statistical procedures

Here is a case in point where you are going to need some statistical tests to discover whether your results are telling you what y ou want to know.

You hope that: there are no differences so far as SEX, SEM, SWOCHST are concerned in the test scores

there are great differences so far as GROUP is concerned in the test scores

i.e. only the course COMPREHENSION has led to an increase in test scores.

YOU NEED THE PROCEDURES _____ (for variables with two categories)

and _____ (for variables with more than two categ.)

(TURN PAGE UPSIDE DOWN FOR THE ANSWER, but think about it for a moment first!)

STATISTICS	ALL
BREAKDOWN	TABLES = CL1 TO TOTAL2 BY SEM SWOCHST
T-TEST	GROUPS = GROUP (1 2) SEX (1 2)/ VARIABLES = CL1 TO TOTAL2

1 16